

Course Material

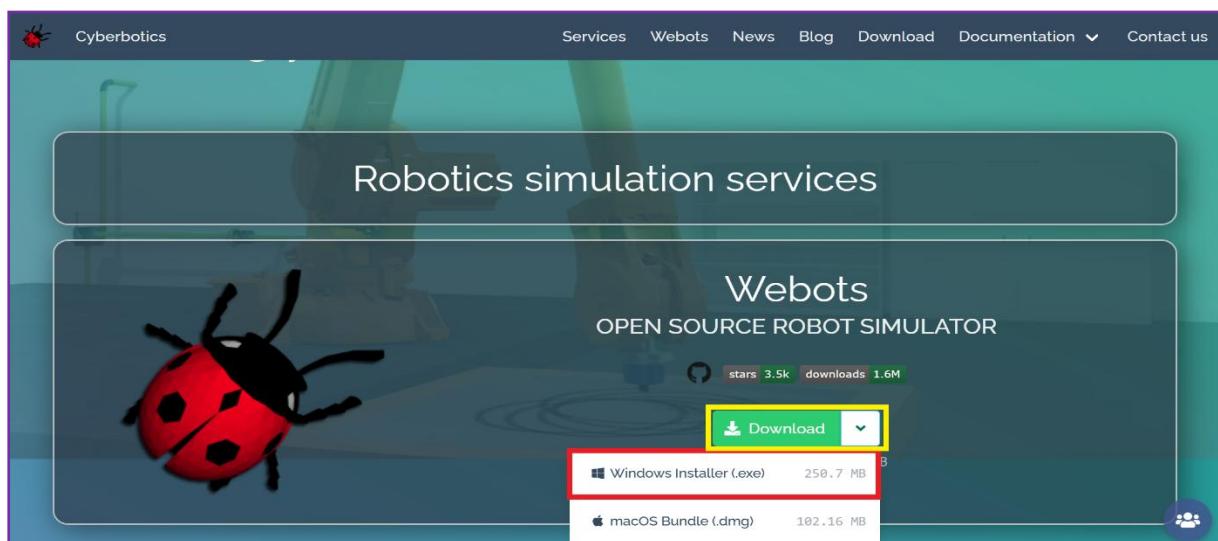
Webots

Table of Contents

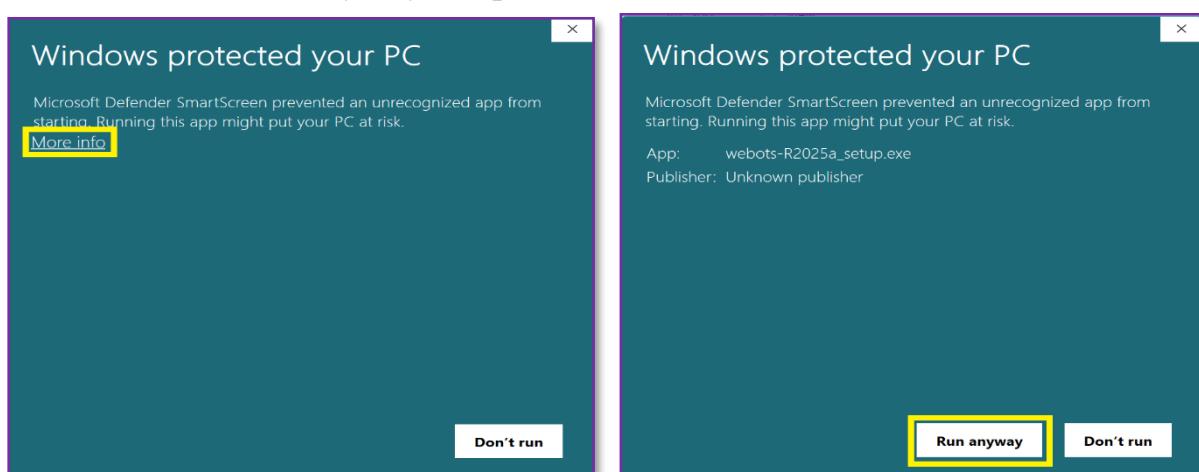
Installation.....	1
Basic Navigation Controls	2
Webots – Add a new object.....	2
Webots – New Robot Controller.....	5

Installation

1. Go to <https://cyberbotics.com/>
2. Click on "Download" → Select "Windows Installer (.exe)".



3. There will be a "warning pop-up" during installation. Click on "More info" and select "Run anyway" to proceed with the installation.

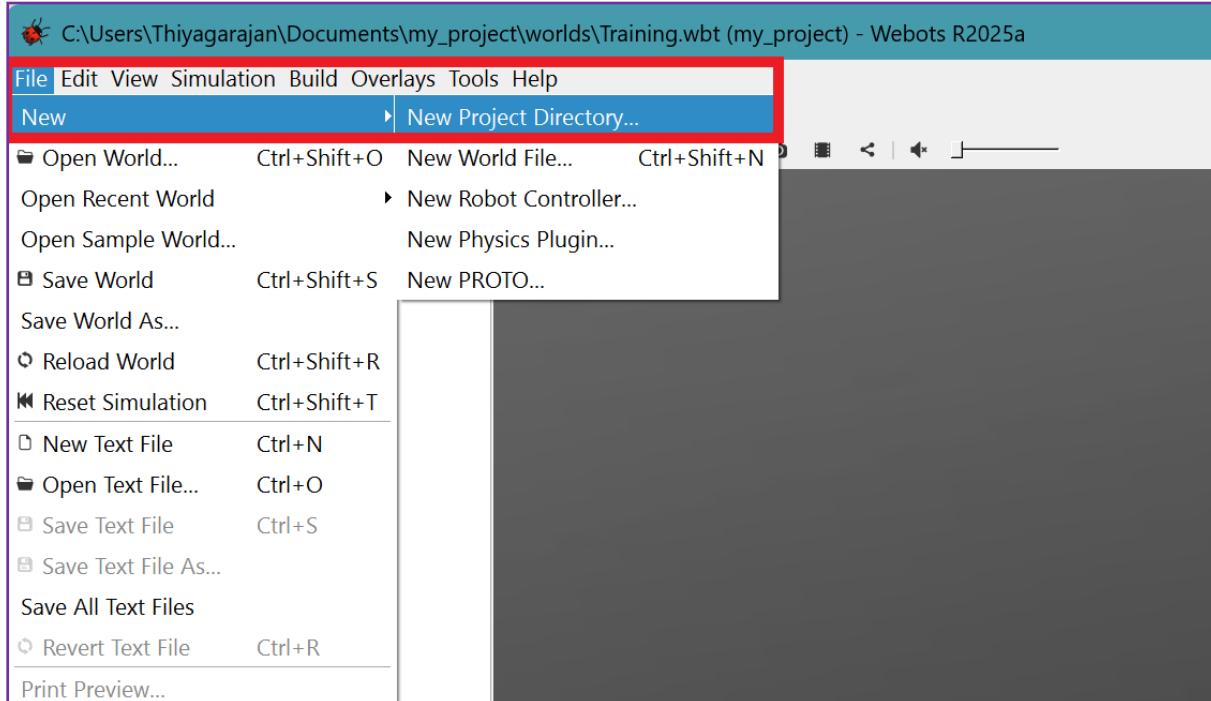


Basic Navigation Controls

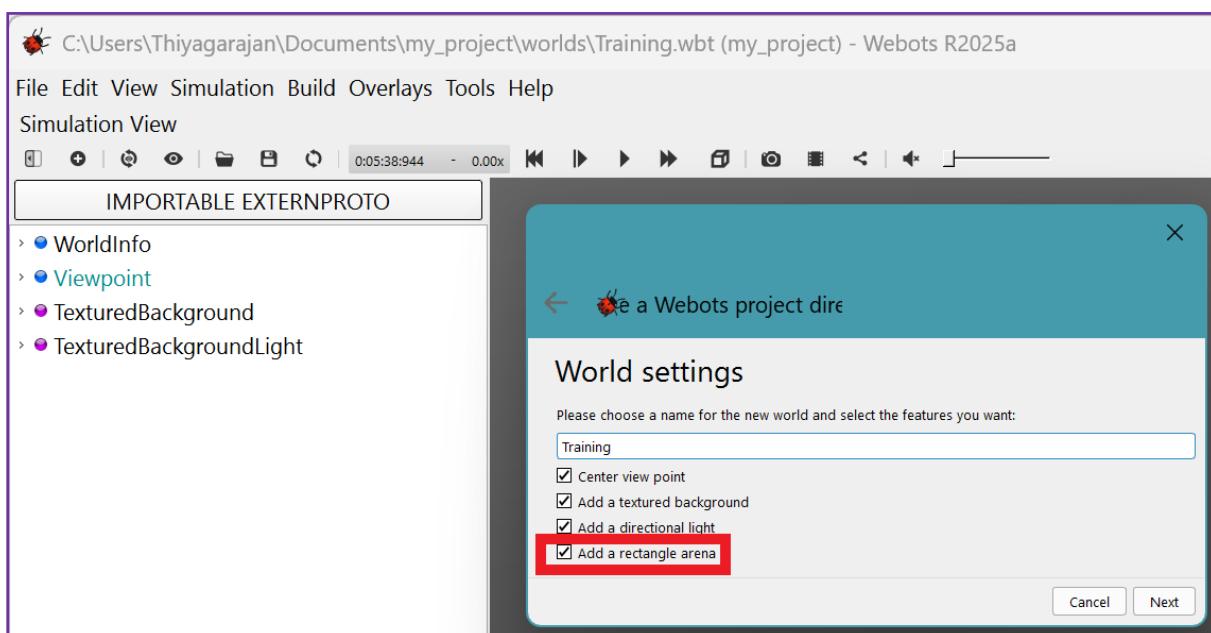
- **Zoom** → Use the **mouse scroll** to zoom in and out.
- **Free Rotate** → Hold **left mouse button** to freely rotate the view.
- **Pan** → Hold **right mouse click** to pan across the workspace.

Webots – Add a new object

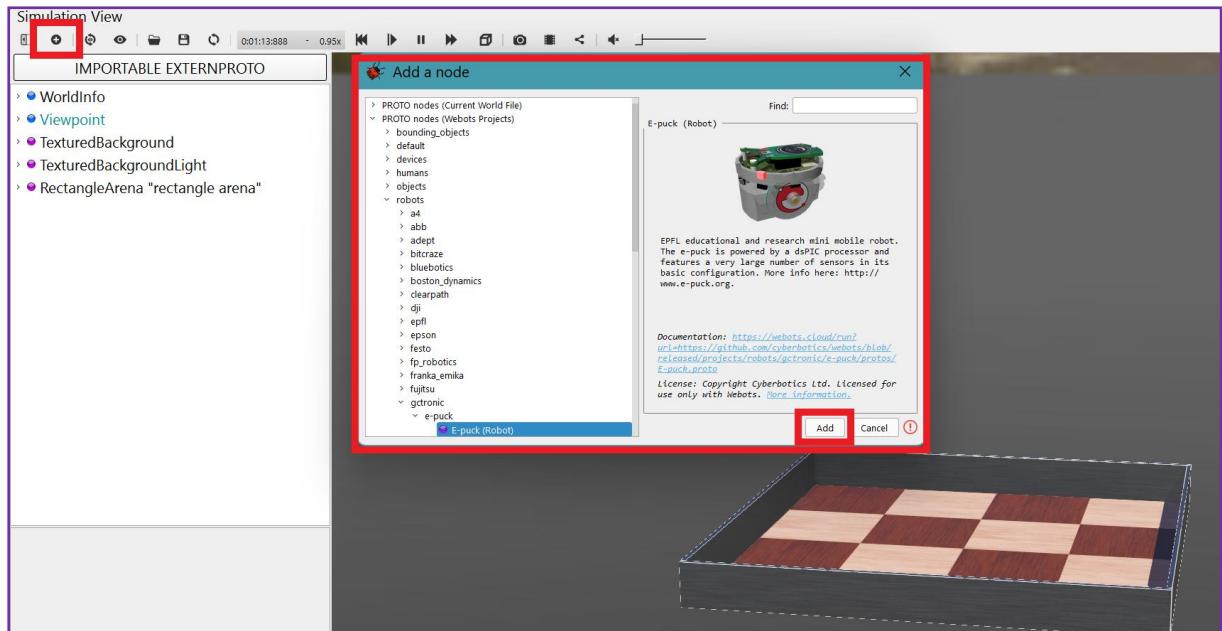
1. Once the application is launched, click on "**File** → **New** → **New Project Directory**".



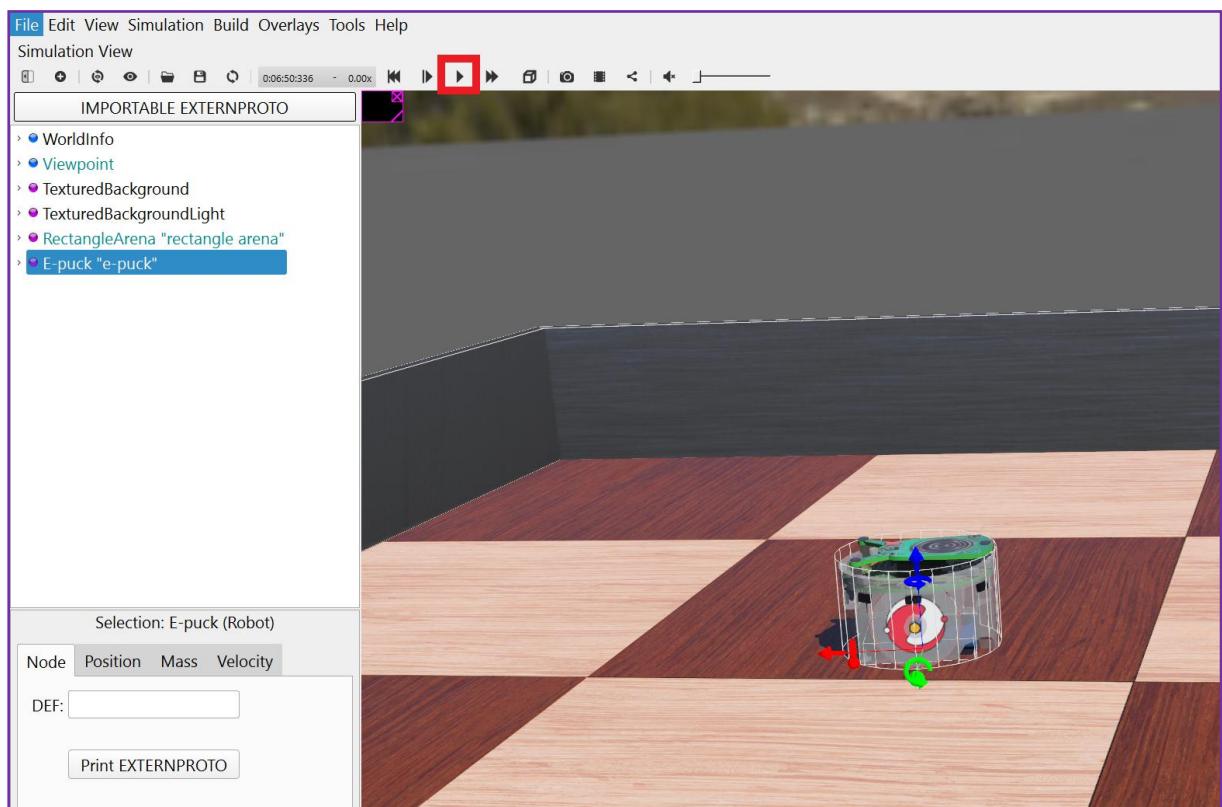
2. While creating the directory, enable "**Add a rectangle arena**" for a rectangle-defined boundary.



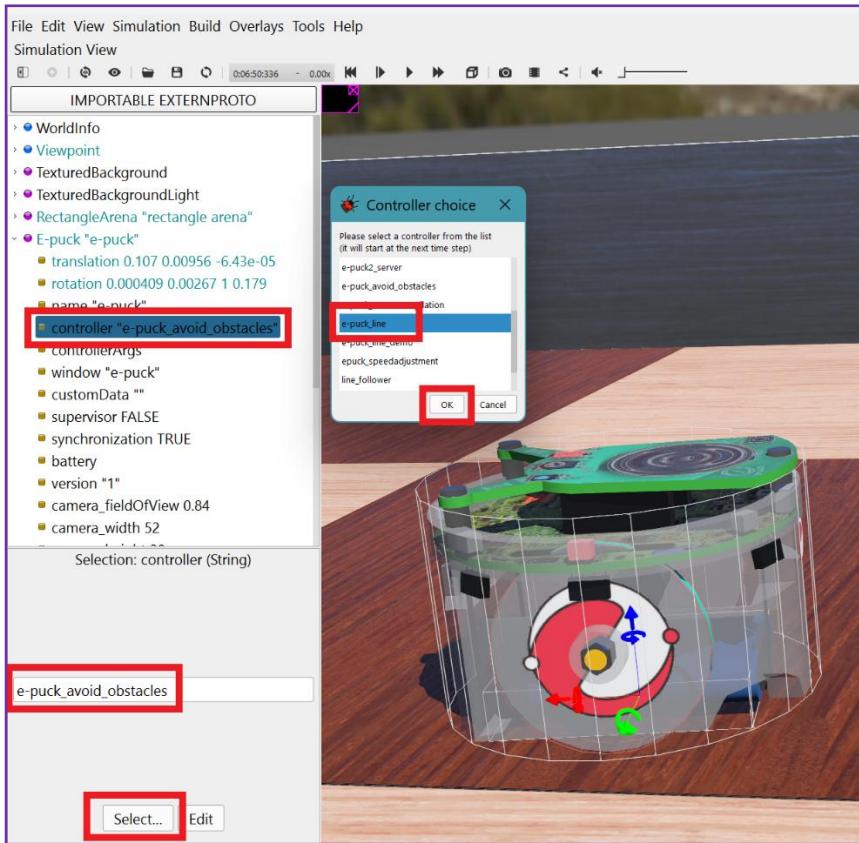
3. Click on "Add a new object" and select "PROTO nodes (Webots Projects) → robots → gctronic → e-puck → E-puck (Robot)". Click on “Add” to add the robot to your workspace.



4. The robot chosen will be added to your workspace. We can play the programmed controller simulation of this robot by clicking on the "Play" button at the top.



5. Different types of **controllers** are available to choose from for **simulation**. Click on "Select" to change the **controller type**. Click on "OK" to simulate the new controller.



6. Click on "Edit" instead to view the **Webots Code** text editor in **C language**.

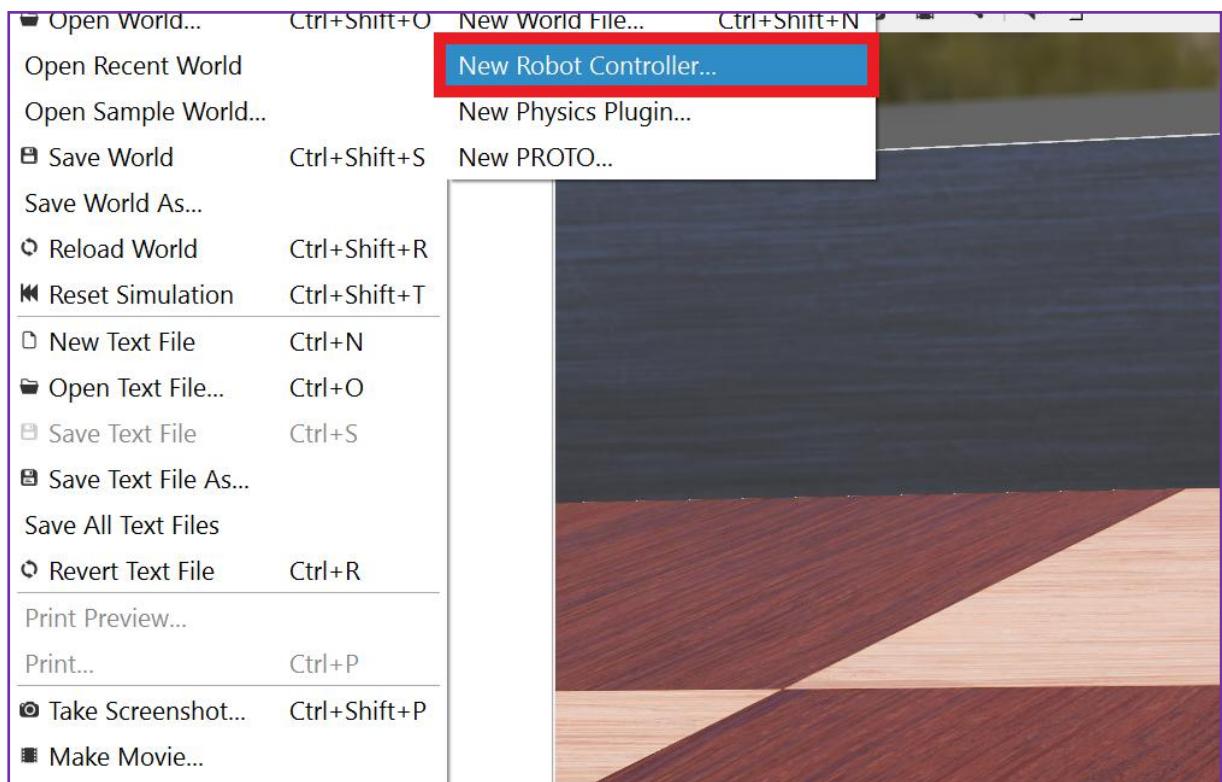
```

C:\Program Files\Webots\projects\robots\gctronic\epuck\controllers\epuck_line\epuck_line.c
52 #include <webots/robot.h>
53
54 // Global defines
55 #define TRUE 1
56 #define FALSE 0
57 #define NO_SIDE -1
58 #define LEFT 0
59 #define RIGHT 1
60 #define WHITE 0
61 #define BLACK 1
62 #define SIMULATION 0 // for wb_robot_get_mode() function
63 #define REALITY 2 // for wb_robot_get_mode() function
64 #define TIME_STEP 32 // [ms]
65
66 // 8 IR proximity sensors
67 #define NB_DIST_SENS 8
68 #define PS_RIGHT_00 0
69 #define PS_RIGHT_45 1
70 #define PS_RIGHT_90 2
71 #define PS_RIGHT_REAR 3
72 #define PS_LEFT_REAR 4
73 #define PS_LEFT_90 5
74 #define PS_LEFT_45 6
75 #define PS_LEFT_00 7
76 #define PS_OFFSET_SIMULATION[NB_DIST_SENS] {300, 300, 300, 300, 300, 300, 300, 300}
77 int ps_value[NB_DIST_SENS] = {0, 0, 0, 0, 0, 0, 0, 0};
78 const int PS_OFFSET_REALITY[NB_DIST_SENS] = {480, 170, 320, 500, 600, 680, 210, 640};
79 // *** TO BE ADAPTED TO YOUR ROBOT ***
80 const int GS_OFFSET_REALITY[NB_GROUND_SENS] = {100, 100, 100, 100, 100, 100, 100, 100};
81
82 // 3 IR ground color sensors
83 #define NB_GROUND_SENS 3
84 #define GS_WHITE 900
85 #define GS_LEFT 0
86 #define GS_CENTER 1

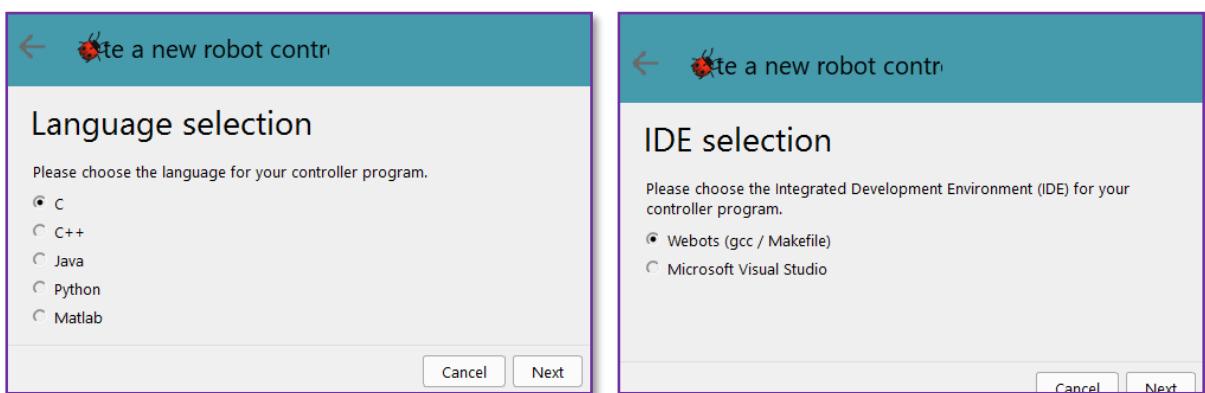
```

Webots – New Robot Controller

1. Click on "File → New → New Robot Controller".



2. Choose the following options: "**Programming language for controller**", "**IDE selection (default)**", and provide a name for your controller. Click on "**Finish**" once done.



The image contains two side-by-side screenshots of a 'Create a new robot controller' wizard. Both screenshots have a teal header bar with a back arrow and the text 'Create a new robot controller'.

- Left Screenshot (Language selection):** The title is 'Language selection'. It asks 'Please choose the language for your controller program.' with a radio button next to 'C' which is selected. Other options include C++, Java, Python, and Matlab. At the bottom are 'Cancel' and 'Next' buttons.
- Right Screenshot (IDE selection):** The title is 'IDE selection'. It asks 'Please choose the Integrated Development Environment (IDE) for your controller program.' with a radio button next to 'Webots (gcc / Makefile)' which is selected. Other options include Microsoft Visual Studio. At the bottom are 'Cancel' and 'Next' buttons.

3. Text Editor in C program will be opened in Webots workspace. Go to [Cyberbotics Tutorial](#) and scroll down to "**Hands-on #10**". Copy the C code and paste it into the Webots text editor. Click on "**Build the current project**" to build the controller.

n Build Overlays Tools Help

0:06:50:336 - 0.00x

TURNPROTO

light angle arena"



C:\Users\Thiyagarajan\Documents\my_project\controllers\my_controller\my_controller.c

```

1 /*
2 * File:           my_controller.c
3 * Date:          my_controller.c
4 * Description:  my_controller.c
5 * Author:        my_controller.c
6 * Modifications: my_controller.c
7 */
8
9 /*
10 * You may need to add include files like <webots/distance_sensor.h>, etc.
11 */
12 */
13 #include <webots/robot.h>
14
15 /*
16 * You may want to add macros here.
17 */
18 #define TIME_STEP 64
19
20 /*
21 * This is the main program.
22 * The arguments of the main function can be specified by the
23 * "controllerArgs" field of the Robot node
24 */
25 int main(int argc, char **argv) {
26     /* necessary to initialize webots stuff */
27     wb_robot_init();
28
29     /*
30     * You should declare here WbDeviceTag variables for storing
31     * robot devices like this:
32     * WbDeviceTag my_sensor = wb_robot_get_device("my_sensor");
33     * WbDeviceTag my_actuator = wb_robot_get_device("my_actuator");
34     */
35 }
```

Cyberbotics

Services News Documentation Contact us

Foreword

Thanks

Installing Webots

Building Webots

Upgrading Webots

Getting Started with Webots

Sample Webots Applications

Programming Language Setup

Development Environments

Programming Fundamentals

Web Interface

Web Server

Tutorials

- Tutorial 1: Your First Simulation in Webots (30 Minutes)**
- Tutorial 2: Modification of the Environment (30 Minutes)**

Hands-on #10: Modify the controller program as shown below, recompile it and run it.

```

#include <webots/robot.h>
// Added a new include file
#include <webots/motor.h>

#define TIME_STEP 64
#define MAX_SPEED 6.28

int main(int argc, char **argv) {
    wb_robot_init();

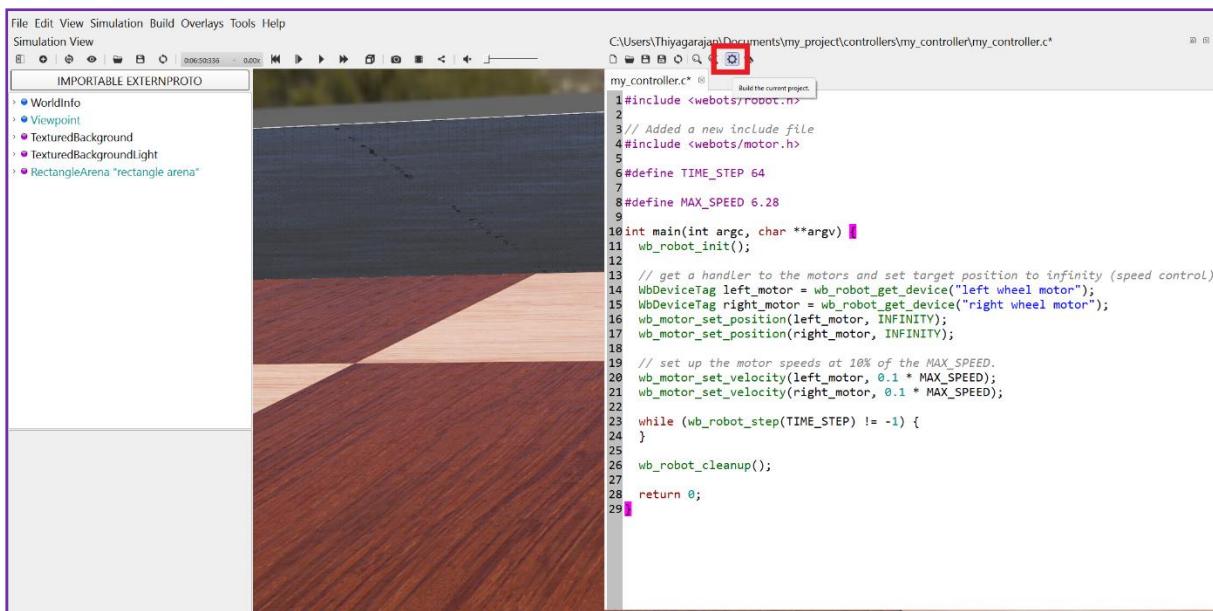
    // get a handler to the motors and set target position to infinity (speed control)
    WbDeviceTag left_motor = wb_robot_get_device("left wheel motor");
    WbDeviceTag right_motor = wb_robot_get_device("right wheel motor");
    wb_motor_set_position(left_motor, INFINITY);
    wb_motor_set_position(right_motor, INFINITY);

    // set up the motor speeds at 10% of the MAX_SPEED.
    wb_motor_set_velocity(left_motor, 0.1 * MAX_SPEED);
    wb_motor_set_velocity(right_motor, 0.1 * MAX_SPEED);

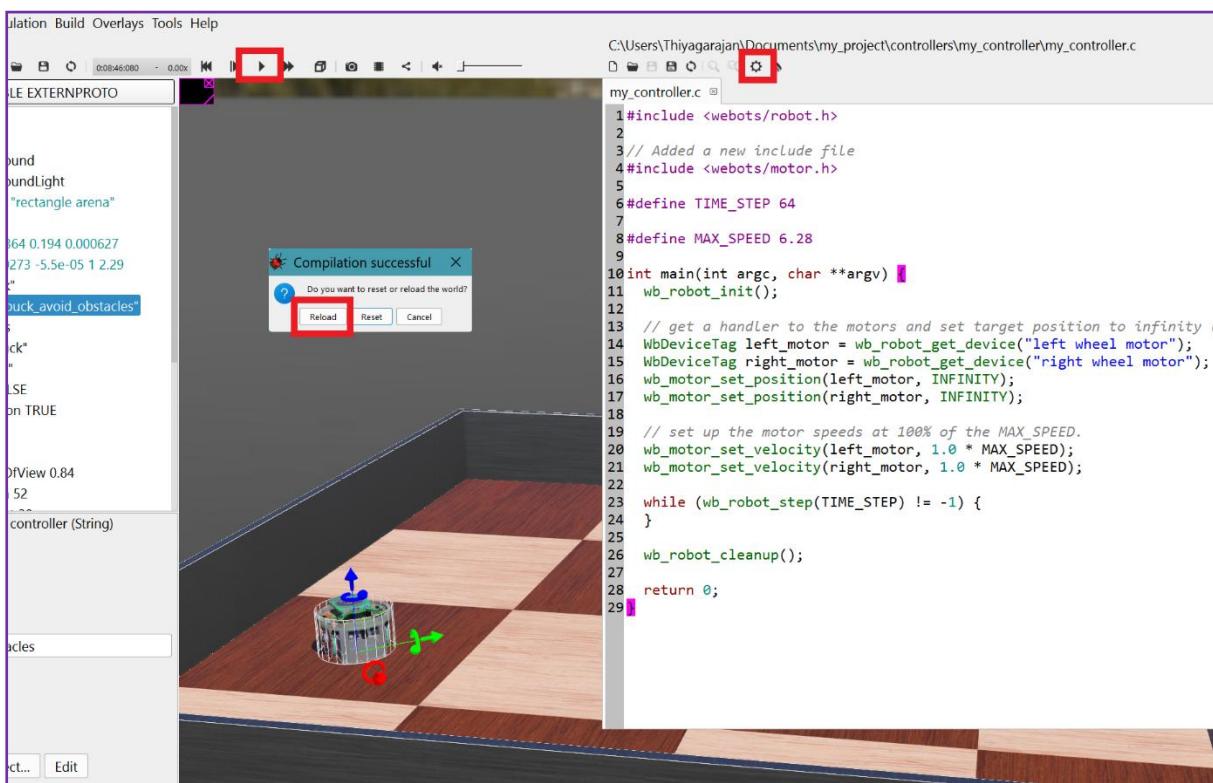
    while (wb_robot_step(TIME_STEP) != -1) {
    }

    wb_robot_cleanup();
}

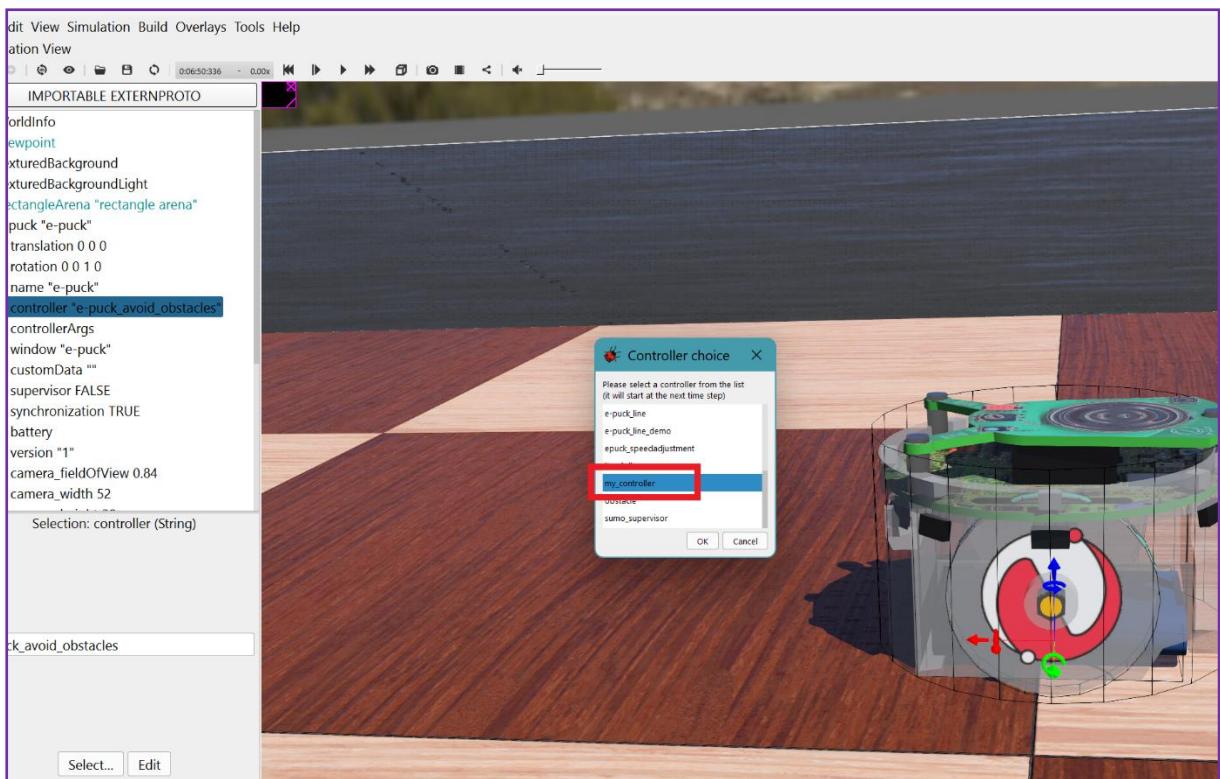
return 0;
```



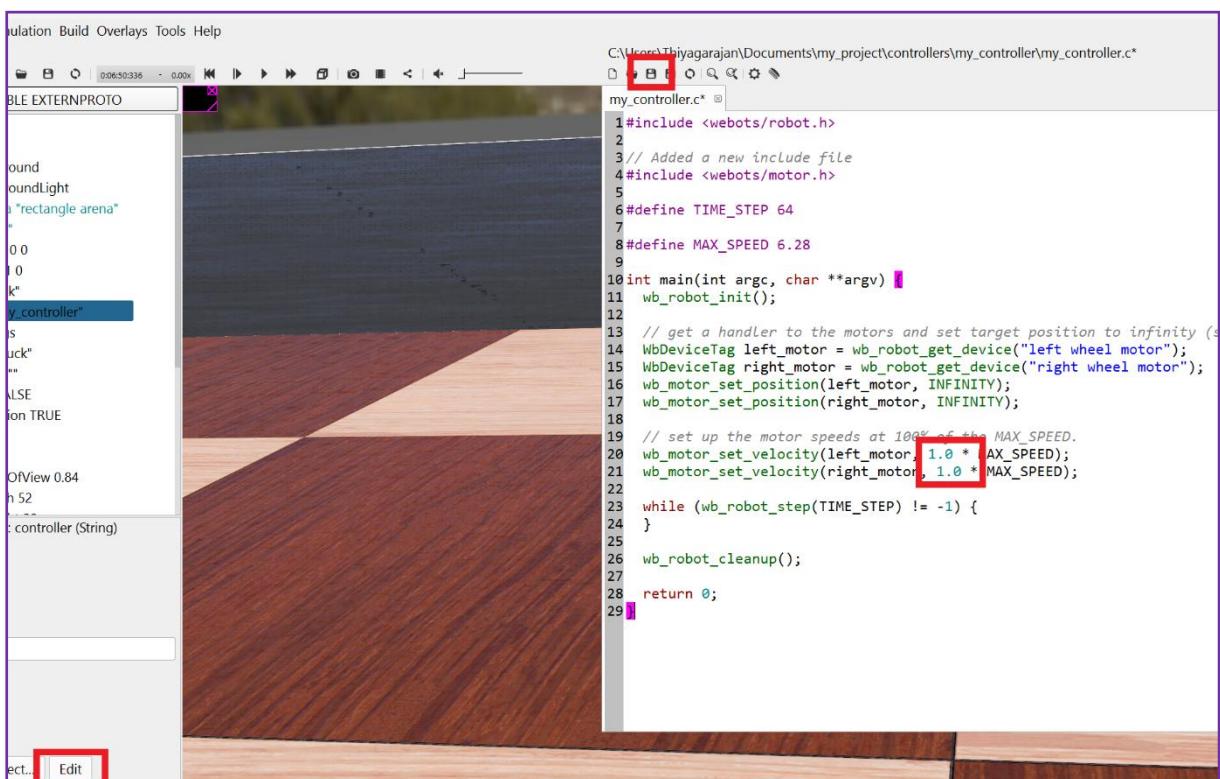
- Click on "Build the current project" and "Reload" the world for the controller. Play the simulation to see the robot moving at a higher speed.



- Now add the robot under "PROTO nodes (Webots Projects) → robots → gctronic → e-puck → E-puck (Robot)". Click on "Add" to add the robot to your workspace.
- Change the controller of the robot to the new one created.



7. Click on "Edit" to change the velocity of motors to **1.0** (i.e., 100% at full velocity) and save the code.



8. Click on "**Build the current project**" and "**Reload**" the world for the controller. Play the simulation to see the robot moving at a higher speed.

